Trust-based Facilitator: Handling Word-of-mouth Trust for Agent-based E-Commerce

Chihiro Ono and Satoshi Nishiyama

KDDI R&D Laboratories Inc.
2-1-15 Ohara Kamifukuoka-shi Saitama, 356-8502, Japan
{ono, tomo}@kddilabs.jp

Keesoo Kim, Boyd C. Paulson, Jr., Mark Cutkosky and Charles J. Petrie Jr.

Stanford University
350 Serra Mall, Stanford University, Stanford, CA 94305
{kskim, Paulson}@stanford.edu, cutkosky@cdr.stanford.edu, and petrie@snrc.stanford.edu

This paper proposes a facilitator which finds capable and trustworthy partners on behalf of client users (agents), which helps agents form and maintain e-partnerships for electronic commerce. Unlike existing capability-based facilitators or matchmakers, the facilitator collects and maintains private "word-of-mouth" trust information as well as capabilities from each agent and uses the information for personalized trust-based facilitation for each agent, which is performed through the facilitation protocols and trust propagation mechanism. Compared to other existing trust mechanisms, the characteristics of trust which this facilitator handles are personalized-collaborative-subjective-qualitative-private. The facilitator is implemented as a JATLite multi-agent system and a FIPA-OS based multi-agent system, and is evaluated in terms of the complexity and characteristics. The example of usage is shown in the area of construction supply-chain coordination.

Keywords: Middle agents, facilitator, trust, e-commerce, KQML

1. Introduction

Currently, online communities where electronic commerce and electronic collaboration are carried out are rapidly expanding along with the growth of the Internet. In these communities, there may be negotiation among automated software programs, called agents. For instance, at auction services, many sellers create auctions for various kinds of goods and many potential buyers bid for goods by following auction protocols. In construction projects, subcontractors negotiate schedules and tasks with general contractors.

In on-line communities for electronic commerce and electronic collaboration, establishing partnerships with which participants (agents) can interact or trade with each other, which we call e-partnerships, is crucial to many applications, such as online auctions and project coordination in various industries. In these cases, agents must have a mechanism for establishing and maintaining partnerships of personally trusted agents, which is based on private word-of-mouth trust information. Also, the partnerships must be dynamic and able to be formed rapidly as application needs dictate, and agents must be able to join the partnerships or be rejected as appropriate.

Our understanding and assumptions of on-line communities are: 1) There are many participant agents and most of them do not know each other. 2) Agents join or leave the community very often. 3) Agents want to keep their opinions of other agents secret. Under those assumptions, it is very important to dynamically find a group of appropriate partners to negotiate with out of a large number of potential partners. This is because, at auction services for example, sellers or auctioneers have to notify potential buyers about the creation of new auctions or, in construction projects, subcontractors have to find potential partners with them to negotiate tasks or schedules.

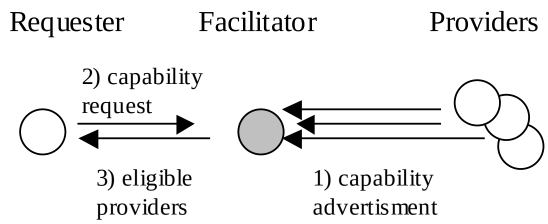

Figure 1. General Mechanism of Facilitator

So far, for the task of finding partners, a concept of facilitator and matchmaker has been proposed. Figure 1 shows a general form of facilitation (Sycara, 1999)
First, provider agents advertise their capabilities to a facilitator agent and the facilitator stores these advertisements. When a requester asks a facilitator whether it knows of providers with the desired capabilities, the facilitator matches the request against the stored advertisements and returns the result, a subset of stored advertisements.
There are several standards which have facilitator-like servers for making dynamic e-partnerships. The Knowledge Query and Manipulation Language (KQML) (Labrou, 1997), proposed as a standard for an agent communication language, assumes the existence of a facilitator and several protocols for facilitation are defined: *broker*, *recommend*, *recruit*, and *subscribe*. The Foundation for Intelligent Physical Agents (FIPA) [6], a standardization body for agent-related technologies such as an agent communication language and agent management, also has a facilitator called Directory Facilitator (DF). The Common Object Request Broker Architecture (CORBA)[3], a standard for developing large-scale distributed object-oriented applications, also has a facilitation server called TRADER. Jini [7], an architecture for developing Java-based distributed applications proposed by SUN Microsystems, also has a facilitation server called Lookup Server. In addition to these standards, there also exist some implementations, such as Matchmaker by CMU (Sycara, 1999) and Kasbah by MIT (Chavez and Maes, 1996).
Unfortunately, however, all of them perform facilitation based only on the registered capabilities of service provider agents and are not sufficient for making e-partnerships under our assumptions. This is because requesters do not want to deal with bad providers. On the other hand, providers also do not want to deal with bad requesters. It is then necessary to filter and rank requests and responses according to trustworthiness for both requesters and providers.
When we think of trust information in the "real world," word-of-mouth information is considered to be very important. There are many quotations on the value of word-of-mouth: "The best prospect is the client who has already dealt with you. The second best is the one referred to by a client who has dealt with you previously. The third best is the one referred to you by another trusted professional or friend" (Marilyn Jennings) [2] "Forget about market surveys and analyst reports. Word of mouth is probably the most powerful form of communication in the business world. It can either hurt a company's reputation or.." (Regis McKenna and others) [2.] Considering this, using word-of-mouth, private trust information seems to be better than using third-party rating systems such as market surveys.
Thus, this paper proposes a private trust-based facilitator for forming e-partnerships which finds partners based on trustworthiness as well as the capabilities of service provider agents.
Section 2 describes the categorization of trust and the trust model. Section 3 explains the design of the facilitator including protocols and the inside mechanism. Section 4 shows the implementation and Section 5 shows the evaluation. Section 6 gives an example of using the facilitator. Finally, we conclude the paper and discuss future work in Section 7.

2. Trust –based Facilitation

"Trust" information, which we try to make use of in facilitation, has been defined differently (Gambetta, 1990: Castelfranchi and Falcone, 1998) and used differently in many applications and services such as rating systems and reputation systems. In this section, we categorize characteristics of trust.

2.1 Trust for e-partnerships

Roughly, we define trust as a general factor for deciding whether or not the facilitator can introduce the agents, as is shown in Zolin's definition (Zolin, 2000): "Trust is the deciding factor in a social process that results in a decision by an individual to accept or reject a risk based on the expectation that another party will perform to the individual's expected performance requirements." And we call trust the value trustworthiness.

As this definition is too vague, however, we define five characteristics of trustworthiness:

1. Commonality of trustworthiness of target agent
   Standardized: same for all participant agents
   Personalized: different from each other
2. Evaluator of target agent
   Authoritative: third-party authority
   Collaborative: participants
3. Objectivity of evaluation
   Objective: based on common criteria
   Subjective: based on different criteria
4. Complexity of trustworthiness
   Quantitative: numeric values
   Qualitative: boolean (positive or negative)
5. Disclosures of trust information
   Public: open to public
   Private: closed to public

Based on this characterization, existing applications are categorized as shown in Figure 2.

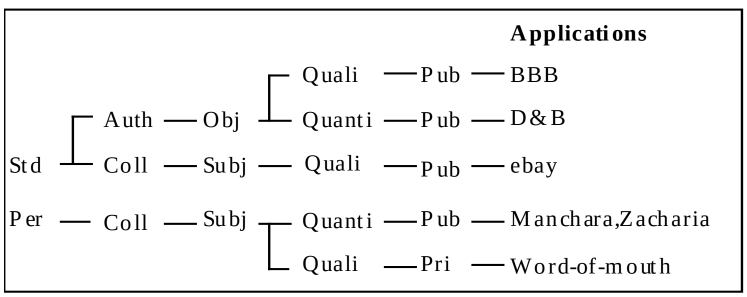

Figure 2. Categorization of applications using trust information

The Better Business Bureau (BBB) [1] and Dun & Bradstreet (D&B)[4] rate companies and provide information to those who inquire about inquired companies, which include trustworthiness, management, profit and so on. BBB has more than 8000 member companies and D&B has rating information for 58 million companies. Both the BBB and D&B evaluate companies by themselves (Authoritative) based on a certain criteria (Objective) and provide a common trustworthiness of target agents (Standardized). For the complexity of trustworthiness, BBB provides a Boolean rating. On the other hand, D&B provides numeric ratings of companies.

eBay[5] runs an auction site to sell and buy various goods, which has more than 10 million members. At eBay, sellers and buyers can check rating scores of potential partners before trading as they evaluate each other by providing feedback after their trades, which means eBay belongs to Collaborative and Subjective. Also, for rating, it belongs to Standardized, Qualitative and Public.

2.2 Handling Word-of-mouth Trust

For taking advantage of word-of-mouth trust information and existing facilitators, which collect capabilities registered by provider agents and perform facilitation based on that information, we propose a facilitator which collects trust information from participants as well as capabilities, and uses both of them for facilitation.

As shown in Figure 2, word-of-mouth trust information is categorized into personalized-collaborative-subjective. In addition to these characteristics, for handling word-of-mouth trust, three important characteristics: transitivity, capability-dependency, and symmetry should be defined. Although these characteristics are discussed differently in various fields, none of them are generally always true. Thus, we define these characteristics as follows by expecting applications to make use of our facilitator by knowing these characteristics:

1. Trustworthiness is transitive. That is, if agent_a directly trusts agent_b and agent_b directly trusts agent_c, agent_a can indirectly trust agent_c. Although there are arguments that trust is not transitive (Christianson, 1996: Josang, 1996), some researches argue that trust is conditionally transitive (Abdul-Rahman, 1997) and some are trying to make use of chain of trust (Zacharia, 1999).
2. Trustworthiness should be capability-dependent. That is, the trustworthiness of one agent concerning car sales could be different from that of the same agent concerning car repair. This is mostly agreed (Abdul-Rahman, 1997: Yu, 1999).
3. Trustworthiness should be asymmetric. That is, even if agent_a trust agent_b, agent_b may distrust agent_a. This is mostly agreed (Abdul-Rahman, 1997).

As for the complexity in section 2.1, we choose "Qualitative" as the calculation should be simple enough to calculate chain of trust for n(n-1) patterns of trustworthiness. Manchala et al. (2000) proposed trust metrics and models for e-commerce by calculating over a chain of numerical trust values (Quantitative) when there is a public intermediary. However, this method is not sufficient for using in trust-based facilitation because the method of building a chain of agents is not mentioned and calculation of numeric values is too complicated. Zacharia et al. (1999) proposed a collaborative reputation mechanism between source and target agents. However, this method also is not sufficient as the calculation used here is overly complex, especially for the calculation of numeric values (Quantitative) and all the paths, including those unused.

2.3 Representation of trustworthiness

The way of representing trustworthiness has the following types based on the requirements described in Section 2.2.

First, cases in which an agent evaluates the target agent directly based on its previous experience with requested capabilities include:

- Direct positive trust (DP): A source agent trusts a target agent directly.
- Direct negative trust (DN): A source agent distrusts a target agent directly.

Second, cases in which an agent evaluates the target agent by using chain of trustworthiness from the source agent to the target agent include:

- Indirect positive trust (IP): A source agent trusts a target agent indirectly.
- Indirect negative trust (IN): A source agent distrusts a target agent indirectly.

Finally, the case in which an agent has no information about the target agent includes:

- Unknown (UN): A source agent cannot decide whether it can trust or distrust a target agent.

Thus, trustworthiness can be represented by any of five types and they are kept in n*n table for each capability, shown in Figure 3. In the figure, agent_a directly trusts agent_e, but agent_e distrusts agent_a.

Trust Table for capability_X						
	AA	AB	AC	AD	AE	AF
AgentA		DP	IP	IP	DP	UN
AgentB	DP		DP	UN	DP	UN
AgentC	IN	DN		DN	DP	UN
AgentD	IP	DP	IP		IP	DP
AgentE	DN	IP	DP	DP		UN
AgentF	UN	IP	UN	DP	UN	

Figure 3. Example of trust table

3. Facilitator Design

Based on the policy discussed above, a facilitator collects capabilities and trust information from participant agents and maintains this information for each capability and uses them in facilitation. There is an approach other than having a facilitator for finding partners, distributed way, with which each participant agent keeps its information by itself and exchanges it with each other. However, using facilitator is better because keeping opinions of other agents secret and maintaining a large amount of capability/trust information in distributed way are very complicated. This section describes a protocol for using a facilitator. In addition, registration, request method, message format and propagation mechanism of trustworthiness are also presented.

3.1 Facilitation Protocols

For communication among requesters, providers and a facilitator, we can use protocols for agent communication such as KQML (Labrou, 1997) and FIPA-ACL[6]. Here, we explain the case of using KQML which provides protocols for facilitation: *broker*, *recruit*, *recommend*, and *subscribe* for requests, as shown in Figure 4.

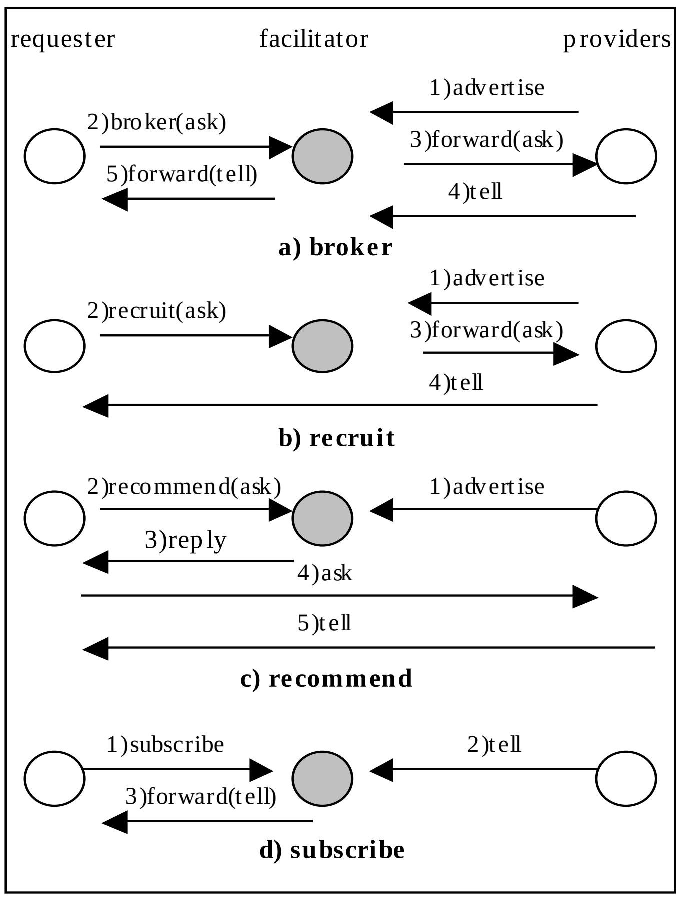

Figure 4. Facilitation protocols

```
Message ::= <msg_type>"("<cap_info>")" [ "("<trust_info>")" ]

<msg_type> ::= REGISTRATION | QUESTION | ANSWER

<cap_info> ::= <context_type>"("<condition>*") ("<extract_term> ") ("<result>* ")"

<trust_info> ::= [<option>] ( "("<agent_name> <trust_value> ")" )

<option> ::= DIRECT_ONLY | POSITIVE | NOT_NEG | ALL
```

Figure 5. Message format

Both capabilities and trustworthiness are described in the "content" parameter of KQML. Figure 5 shows a message format of the content parameter. A “Message” comprises message type, capability information and trust information. A message type includes REGISTRATION, QUESTION and ANSWER. Capability information includes context type, condition, extract terms and result. “Context type” is a name of capability, and we assume global namespaces, in which all the participants have a common vocabulary about their capabilities and attributes. From them, only the context type is parsed at the facilitator and the rest are parsed either at requesters or providers. Trust information includes option and pairs of agent name and trustworthiness. “Option” means degree of using trustworthiness on facilitation, and it can be requested by both requesters and providers. The choices are:

- DP only
- DP and IP
- Not negative (that is, DP and IP and UN)
- All

```
a) advertise
  :sender agent_a :receiver facilitator1
  :reply-with label1
  :content (REGISTRATION(car_sales()()())
            ((agent_b P)(agent_c N)))
b) broker-all
  :sender agent_b :receiver facilitator1 :reply-with label1
  :content (QUESTION(car_sales((>year 1997)(<price
            9000))(Make Price)())(POSITIVE (agent_c P)))

c)tell
  :sender facilitator1:receiver agent_a
  :in-reply-to label1
  :content (ANSWER(car_sales()())
            ((Toyota 8000)(FORD 7050)))
```

Figure 6. Examples of KQML Message

Figure 6 shows an example of KQML messages which follow “broker” protocol shown in figure 4(a). Here, registration of trustworthiness is carried out by using the "advertise" performative of KQML. For provider agents, registration can be done with registration of capabilities as shown in Figure 6 (a), in which provider agent (agent_a) advertises to the facilitator1 its capability “car_sales” and registers trustworthiness; that means it trusts agent_b and distrusts agent_c. For requester agents, registration of trustworthiness can be done with a request for facilitation as shown in Figure 6 (b), in which agent_b requests for brokering to the facilitator1 with conditions that the year be newer than 1997 and price be less than $9000. This also means the agent expects Make and Price as a result and, for trustworthiness, registers that it trusts agent_c and wants to get only directly trusted agents.

Shown in Figure 6 (c), is an example of an answer from the facilitator1. The results are a Toyota, which costs $8000 and a FORD, which costs $7050. If the facilitator finds more than one answers, it will give all of the answers and let providers choose the best one.

3.2 Trust Propagation Mechanism

Inside the facilitator, filtering potential partners is performed based on requested capabilities and trustworthiness. For maintaining registration in the facilitator, capabilities and trustworthiness are stored in a dynamic database inside the facilitator. In the database, trustworthiness values are stored by an n*n matrix, in which the n is the number of registered agents, for each capability type.
Trustworthiness registered by users could be either DP or DN. When a facilitator receives data, cells of the matrix which remain UN may be converted into IP or IN by calculation. Every time the facilitator receives new data, it recalculates all of IP, IN, UN data. Calculating indirect trust (IP or IN) of one cell, from agent X1, to agent Xn is done by the following two steps:

[Step 1] Finding paths from X1 to Xn based on the following policies:

- An agent can use only the direct trust of other agents.
- From Xj to Xj+1 (1 < j < n-2), only DP can be used.
- From Xn-1 to Xn, both DP and DN can be used.

[Step 2] Tie-breaking if more than one path exists based on certain rules such as:

- Majority decision: firstly score each path based on certain criteria such as the number of hops from the source to the destination, and then compare the total scores of positive and negative paths.
- Risk averse decision: IN if at least one negative path is found.

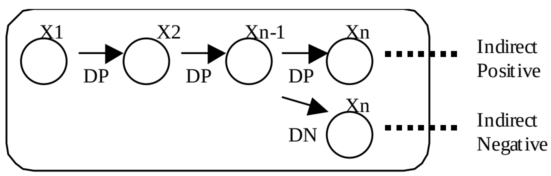

Figure 7. Indirect trust

As far as the computational complexity for calculation is concerned, the worst case is to find all possible paths and compare them, for all cells of the table. In this case, the complexity is:

$$\begin{aligned}&(number\_of\_cells)*(number\_of\_possible\_paths)\\&=(n*n)*(1+(n-2)+(n-2)(n-3)+(n-2)(n-3)..1)\\&=O(n^n)\end{aligned}$$

This method is not feasible as it is NP complete. Thus, to reduce the complexity, following two heuristic techniques are considered:

[Technique 1] Searching within limited length of paths, and
[Technique 2] Finding the shortest path

Technique 1 is to find paths which have length less than the specified length, and to apply tie-breaking is more than one path exists. The complexity of this technique is:

$$\begin{aligned}&(n*n)(1+(n-2)+(n-2)(n-3)+(n-2)..(n-q))\\&=O(n^{(q+1)})\end{aligned}$$

Where q is the specified length.

Thus, feasible complexity can be achieved if q is a small number.

Technique 2 is to find shortest paths and to apply tie-breaking if more than one shortest path exists. The way of finding the shortest paths is as follows:

1) Making k_th (1<k<n) reachable matrixes (RM) which have n*n cells, which show reachability from node i to node j within k steps. Let $V_k(i, j)$ be the value of the (i, j) cell of Kth RM.
   - Making 1st RM,

     $V_1(i, j) = 1$ where i = j, or (i, j) = DP

     otherwise 0
   - Making k+1_th RM

     $V_{k+1}(i, j) = 1 : where \sum_{x=1}^{n} \{V_k(i, x) * V_1(x, j)\} > 0$ otherwise 0
2) Calculating indirect trust by using k_th RM.
   - For each cell (i, j), finding all the nodes e[x] which have direct connection (DP or DN) to node j and calculate the length of shortest paths from node i to each e[x] by using RM.
   - Apply tie-breaking if more than one shortest path exists (majority decision or risk averse decision)

Thus, the complexity of Technique 2 is:

$$
\begin{aligned}
&(Complexity\_of\_making\_nth\_reachable\_matrixes) \\
&+ (number\_of\_cells) * (number\_of\_previous\_nodes) \\
&= (number\_of\_cells) * (calculation\_for\_V_k(i, j)) \\
&+ n^2 * (n-2) \\
&= O(n^4) + n^2 * (n-2) \\
&= O(n^4)
\end{aligned}
$$

Then each of two techniques can be used based on the application domain where the facilitator is applied.

4. Implementation

We have implemented facilitators based on both KQML and FIPA standards. For KQML standard, we have implemented the facilitator as a JATLite-based multi-agent system, developed at Stanford University (Jeon and Petrie, 2000). For FIPA standards, we have implemented the facilitator as a FIPA-OS-based multi-agent system, developed at Emorphia (Poslad, 2000). JATLite is a Java-based platform and consists of a message router for exchanging messages between agents and a template for developing agents which speak the KQML language. All messages are exchanged through the message router, as shown in Figure 8.

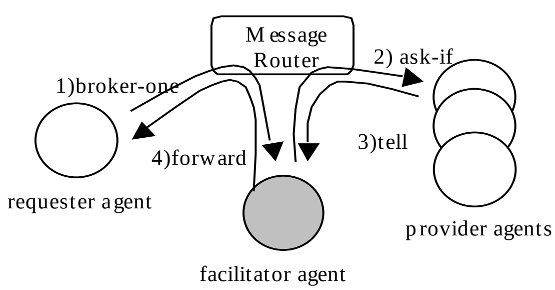

Figure 8. JATLite and Facilitator

Figure 9 shows a GUI of requester agents. In the figure, a request for facilitation with trustworthiness is described in the content parameter at the bottom left of the window and the answer forwarded by the facilitator originating from the provider is described at the bottom right of the window.

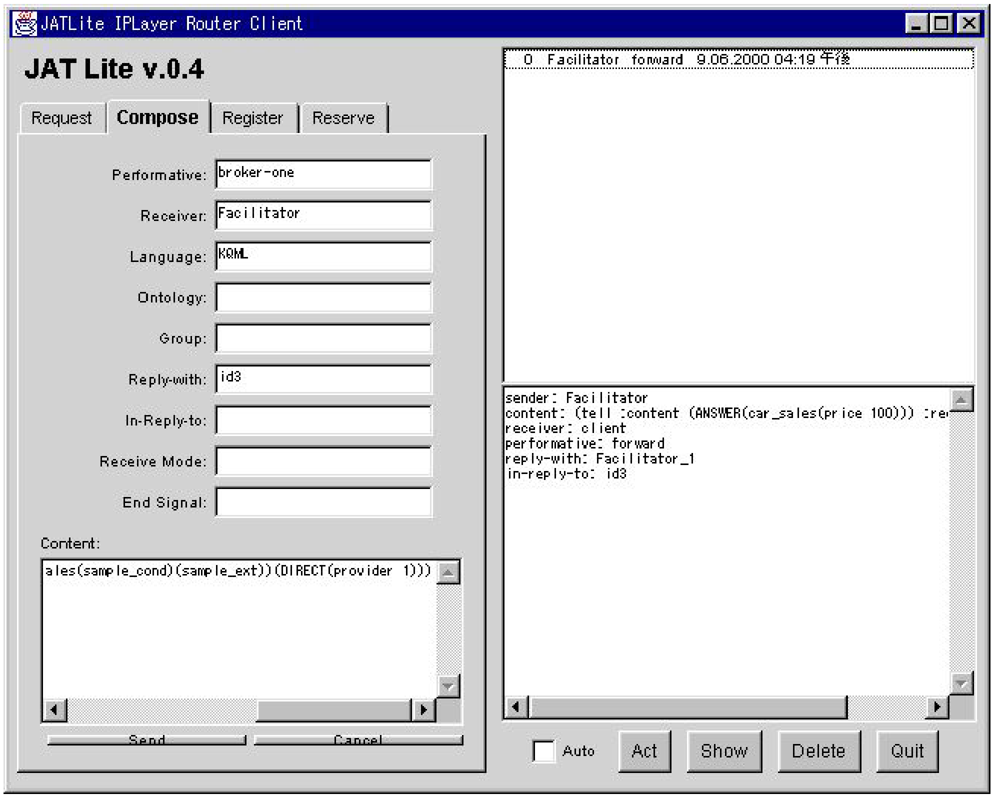

Figure 9. Broker-one request Example

FIPA-OS is also a java based agent platform and consists of an Agent Shell for producing agents which can communicate with each other using FIPA-OS facilities and FIPA-ACL, message and conversation management, and generic agents such as Agent Management System for managing lifecycle of agents and Directory Facilitator for providing yellow page services as shown in Figure 10. We extend Directory Facilitator to support trust based facilitation as well as the capability-based facilitation.

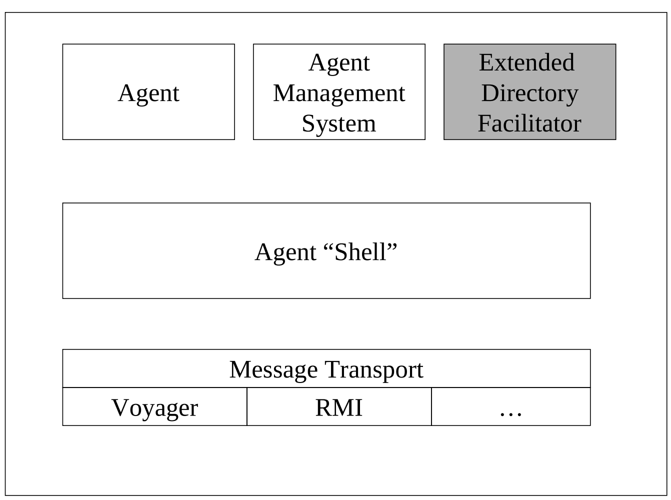

Figure 10. FIPA-OS and Facilitator

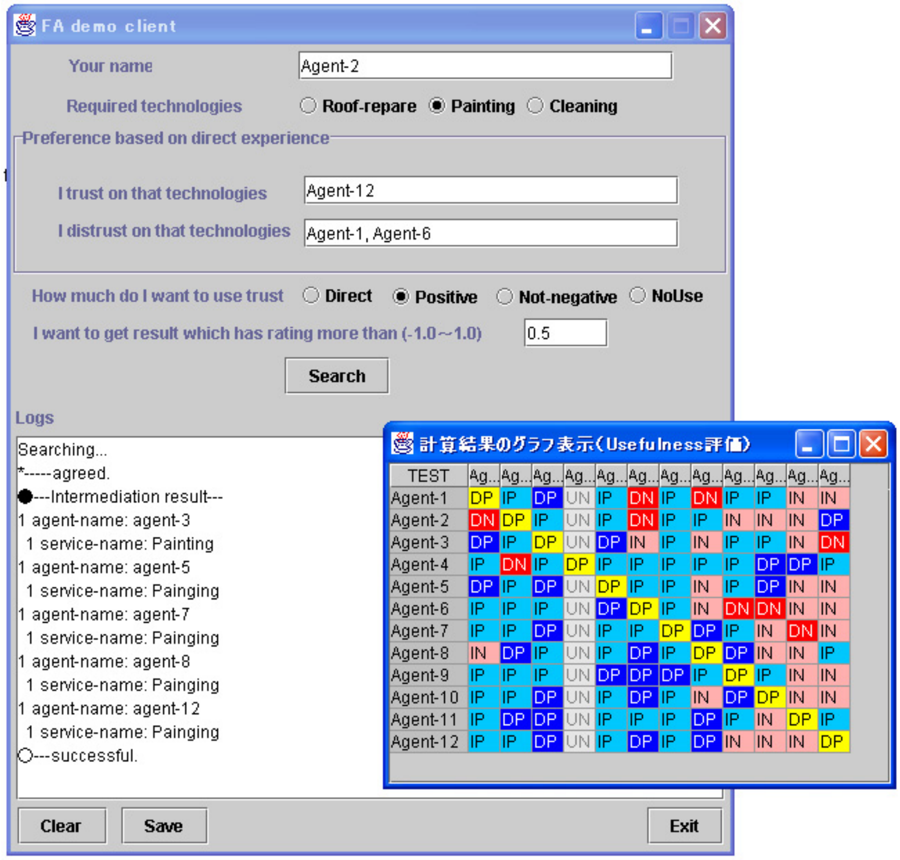

Figure 11. Facilitation example on FIPA-OS based facilitator

Figure 11 shows a GUI of requester agents and the screenshot of trust table. In the figure, agent-2 sends a request for facilitation with trustworthiness saying that it trusts agent-12 and it distrusts agent-1 and agent-6. It also requests the degree of using trust as positive which means that agents with DP and IP can be introduced. The result is described at the bottom left of the window, which say that the agent-3, 5, 7, 8, 12 are introduced.

5. Evaluation

5.1 Performance Characteristics of the Facilitator

Each application which uses trust table has a requirement for the characteristics of the trust table such as 1) complexity of calculation of the trust table, 2) cover ratio of cells which is the ratio of non-UN cells for all cells, and 3) the ratio of IN within indirect trusts. On the other hand, some characteristics of the application which are related to trust table can be observed such as 1) the approximate number of registered agents, 2) the average number of trust information held by each agent, 3) The ratio of DN within direct trust.

Here, for calculating indirect trust, the facilitator can choose several parameters such as heuristic method, tie-breaking rules and the length of paths to be examined. For deciding the parameters, it is very important to know the relationship between the characteristics of the applications and the characteristics of the trust table.

As for the complexity of calculation, it is carried out for updating trust table when the facilitator receives the new trust information. Although the required response time for calculation is not as severe as it is for facilitation requirement, it should achieve the feasible response time. On the other hand, it should be noticed that the facilitator has a quick response with order of constant value for facilitation with trust information where the calculation does not occur.

As for the cover ratio, it is basically affected by the average number of trust information held by agents. In addition, it is also much affected by the length of paths to be examined when heuristic technique 1 in section 3.2 is applied. Thus, we have to take into account of the relationship between the length of paths for reducing the complexity and the cover ratio.

As for the ratio of IN within the indirect trust, we should be careful when we use risk averse tie-breaking rule as most of the indirect cell may become IN when we have many DN. So we have to understand their relationships.

Based on the discussion above, later in this section, we perform the following four evaluations to collect the basic data for making use of how to decide the parameters of trust table in applying specific applications.

1. Relationship between the number of trust information and the cover ratio.
2. Relationship between the number of agents and calculation time.
3. Relationship between the average number of trust information and calculation time.
4. Relationship between the ratio of registered DN and the ratio of IN.

5.2 Settings

The methods used in this evaluation are shown in table 1.

Method	Heuristic applied	path length	Path check	Tie-breaking
A	1	within limited number of paths	All	Majority Decision
B	2	All length	only the shortest path	Majority Decision
C	2	All length	only the shortest path	Risk Averse
D	1 and 2	within limited number of paths	only the shortest path	Majority Decision

Table 1. Methods applied in the evaluation

Parameters used in the evaluation are followings

- The number of registered agents
- The average number of trust info held by agents
- Ratio of DN within direct trusts
- Length of path (For method A and D)

5.3 *Results*

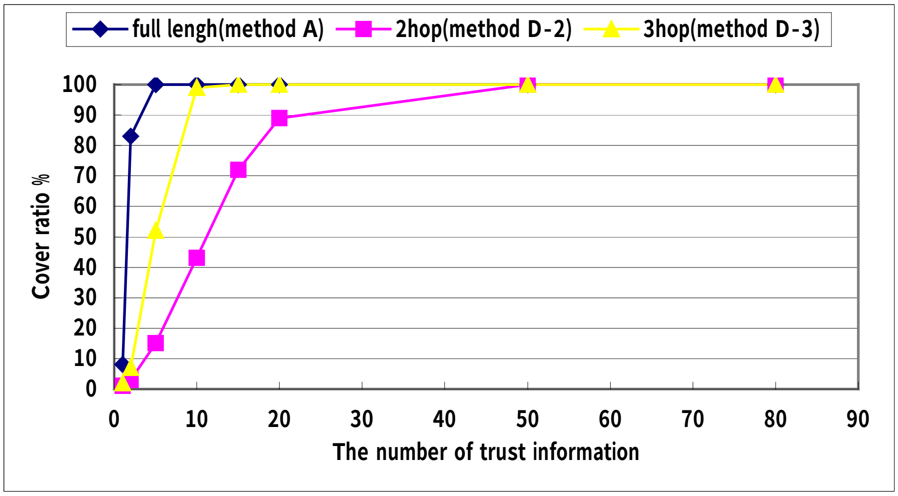

Figure 12. Cover ration against the number of trust information

Figure 12 depicts the relationship between the average number of trust information of each agent and the cover ratio, for each method. Here, three types of the lengths are tested and the number of registered agent is 200 and all the direct information is DP.

All the cells are filled for full length when average number of trust information is 5, which is 2.5% of all registered agents, when the number is 10 (5%) for 3 hops, and when the number is 50 (25%) for 2 hops. When 50% of cover ratio is enough, then the required number decreases to 5 for 3 hops and 10 for 2 hops.

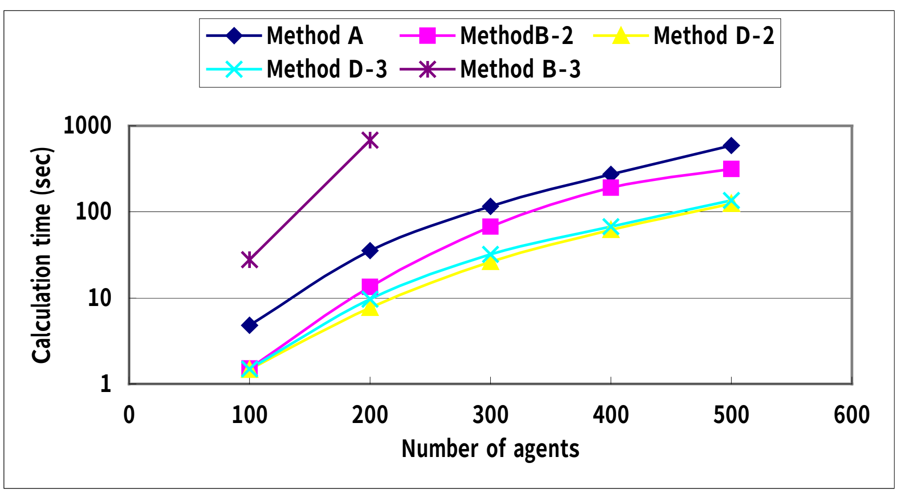

Figure 13. Calculation time against the number of agents

Figure 13 demonstrates the relationship between the number of registered agents and calculation time, for each method. Here, the average number of trust information is 10% of the number of registered agents where cover ratio is 100%.

For method B, it takes much time than method A, which searches all the hops if the number of hops is 3 or more.

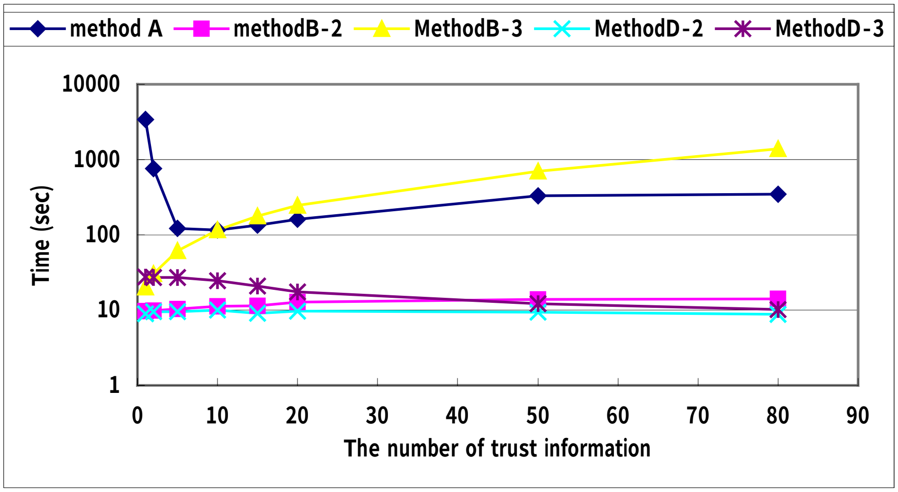

Figure 14. Calculation time against the number of trust information

Figure 14 presents the relationship between the average number of trust information and calculation time for each method. Here, the number of agent is 100.

As for the method B, the calculation time becomes larger as the number of target cells increases. As for method A, the calculation time is large if the cover ratio is less than 100% since the facilitator has to check (n-1) hops for the cell which results in UN.

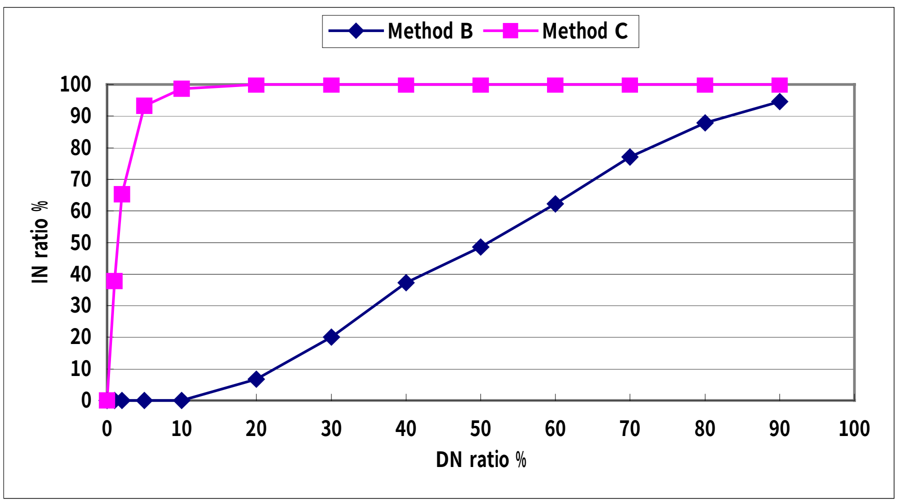

Figure 15. IN ration against DN ratio

Figure 15 shows the relationship between the ratio of DN and the ratio of IN. At method C, which is risk averse type, almost all the indirect cells become IN when the ratio of DN is more than 20%.

Above all, the knowledge lead by these results for deciding parameters are as follows:

- We have to choose the length of the paths examined, by considering the tradeoff between the cover ratio and calculation time based on the application requirements.
- If the number of the agents is not so big like 100 to 1000, which means a closed group, this facilitator works well in terms of the cover ratio.
- If the application allows applying heuristic 1, it is good to use method B or D as it is much faster than method A. Whether applying B or D depends on the required cover ratio of the application.
- As for method B, it takes much time when the cover ratio is less than 100%. Thus, if we cannot expect 100% cover ratio, practically, it is good to use method D and limit the length of paths.

- The facilitator can introduce better partner agents by applying risk averse rule but might not be able to introduce any agents if there are many DNs.

We can set parameters by considering these characteristics.

6. Facilitator Use

Take supply chain coordination at a construction project as an example. Recently, construction projects are carried out by general contractors who get an order and coordinate subcontractors who actually do the work. Consequently, a network for Project Supply-Chain coordination, where negotiation of task and schedule are performed, has been established, as shown in Figure 16.

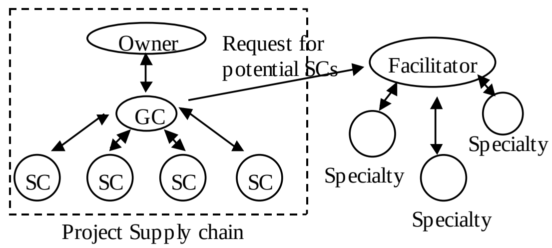

Figure 16. Agents in Project Supply Chain

The project supply-chain coordination requires the collaboration of numerous suppliers and subcontractors. In particular, the degree of collaboration, including the sharing and joint creation of extensive information as well as the sharing of risks and benefits in the face of uncertainty, requires that the collaborators have a degree of mutual trust. The facilitator will help participants to form and maintain mutual trust information through trust-based facilitation for the project supply chain coordination.

The facilitator generally provides participants with opportunities to seek capable and trustworthy partners with whom they want to work. In cases of external changes in construction projects, which are ubiquitous in construction, the participants could seek outside partners to alleviate their losses. The facilitator could provide a longer list of eligible partners through the trust propagation mechanism than current practices where each participant maintains its list of eligible partners respectively.

As an example, suppose there are one general contractor (GC) and five subcontractors (SUBs). Suppose that the GC wants to subcontract some portion of its work -- C3 -- to a selected subcontractor, as shown in Figure 17.

Capability Table

	C1	C2	C3	C4	C5
SUB_A	V			V	
SUB_B		V	V	V	V
SUB_C	V		V		V
SUB_D	V	V	V	V	
SUB_E	V	V	V	V	V

Trust Table for C3

	GC	SA	SB	SC	SD	SE
GC		DP	IP	IP	DP	UN
SUB_A	DP		DP	UN	DP	UN
SUB_B	IN	DN		DN	DP	UN
SUB_C	IP	DP	IP		IP	DP
SUB_D	DN	IP	DP	DP		UN
SUB_E	UN	IP	UN	DP	UN	

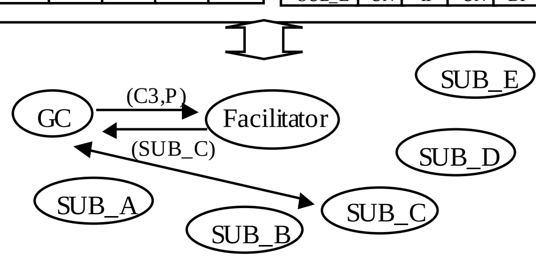

Figure 17. Trust-based facilitation

The GC wants to work with a capable and trustworthy subcontractor for the job. Therefore, the GC wants to ask the Facilitator to provide a list of eligible subcontractors by sending a message, such as QUESTION(C3&POSITIVE), to the Facilitator. The Facilitator checks the Capability Table and finds four capable subcontractors, such as SUB_B, SUB_C, SUB_D, and SUB_E. Among these SUBs, the Facilitator checks the Trust Table and finds three trustworthy SUBs – SUB_ B, SUB_C, and SUB_D – which have POSITIVE trustworthy values for the GC. Before sending a list of these subcontractors to the GC, the facilitator checks the Trust Table again and finds that SUB_B and SUB_D do not want to be introduced to the GC, based on their NEGATIVE trustworthy values to the GC. Therefore, the Facilitator reports only one eligible subcontractor by sending a message, such as ANSWER(SUB_C), to the GC. Then the GC negotiates with SUB_C for the job.

The Facilitator calculates and keeps trustworthiness values when the GC and SUBs register capability and provide DP or IP values for others with whom they have direct experiences. Default UN values will be changed to IP or IN through the Facilitator's trust propagation mechanism. For example, for the GC, the Facilitator tags IP for SUB_B because trustworthy SUB_A trusts SUB_B; IP for SUB_C because trustworthy SUB_D trusts SUB_C; and UN for SUB_E because trustworthy SUB_A and SUB_D have no information about SUB_E. Note that SUB_D does not trust GC even though GC trusts SUB_D. The trustworthy values are subjective for each one. Because of that, SUB_B has IN value to the GC. The unknown values of SUB_E will be changed after the GC evaluates SUB_C because both SUB_C and SUB_E trust each other.

Note that the GC has trusted only SUB_A and SUB_D before this facilitation process. With the aid of the facilitator, the GC will know more trustworthy SUBs than before. The more facilitating process will enrich the value of the facilitator, which means that the facilitator could suggest more eligible partners.

7. Conclusion

We propose a facilitator which finds capable and trustworthy partners on behalf of client users (agents). Unlike existing capability-based facilitators or matchmakers, the facilitator collects and maintains private "word-of-mouth" trust information as well as capabilities from each agent and uses the information for personalized trust-based facilitation for each agent, which is performed through the facilitation protocols and trust propagation mechanism. We believe that this facilitator is the first trust-based facilitator which uses personalized-collaborative-subjective-qualitative-private trust information.

There are some further studies to be improved. First, we can devise a more reliable algorithm for trust propagation in terms of consistency, simplicity and relevancy. We also can add the self-healing mechanisms against malicious agents and support for newcomers to the community.

Notes

1. Better Business Bureau http://www.bbb.org
2. Cafferky M. Free Word-of-mouth Marketing Tips home page. http://www.geocities.com/WallStreet/6246/quate6.html
3. CORBA Specifications http://www.corba.org
4. Dun and Bradstreet http://www.dnb.com
5. eBay http://www.ebay.com
6. FIPA Specifications http://www.fipa.org
7. Jini Specifications http://www.sun.com/jini/

References

Abdul-Rahman, A. and Hailes, S. (1997). "A Distributed Trust Model", ACM New Security Paradigms Workshops

Catelfranchi, C. and Falcone, R. (1998). "Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification," In Proceedings of the 3rd International Conference on Multi Agent Systems, 72-79

Chavez, A. and Maes, P.(1996). " Kasbah: An agent Marketplace for Buying and Selling Goods," In Proceedings of PAAM'96, London, UK

Christianson, B. and Harbison, W.S. (1996). "Why Isn't Trust Transitive?," In Proceedings of Security Protocols International Workshop, University of Cambridge

Gambetta, D. (1990) "Can We Trust Trust?," Making and Breaking Cooperative Relations, Basil Blackwell, Oxford

Jeon, H., Petrie, C. and Cutkosky, M. R. (2000). "JATLite: A Java Agent Infrastructure with Message Routing," IEEE Internet Computing, March-April

Josang, A. (1996). "The Right Type of Trust for Distributed Systems," In Proceedings of New Security Paradigms 96 Workshop

Labrou, Y. and Finin, T. (1997). "A Proposal for a new KQML Specification," TR CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD 21250.

Manchala, D.W. (2000). "E-Commerce Trust Metrics and Models," IEEE Internet Computing, 4(2), 36-44

Sycara, K. et al. (1999). "Matchmaking among Heterogeneous Agents on the Internet," In Proceedings of AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, USA

Yu, B. (2000). "A Social Mechanism of Reputation Management in Electronic Communities," In Proceedings of Cooperative Information Agents, CIA-2000, Boston, MA, USA, 154-165

Zacharia, G. and Maes, P. (1999). "Collaborative Reputation Mechanisms in Electronic Marketplace," In Proceedings of HICSS-32

Zolin, R. et al. (1999). "Building, Maintaining And Repairing Trust In Global AEC Teams". In Proceedings of ICCCBE-VIII, Stanford CA, ASCE, 874-88

Poslad S. et al. (2000). "Open Source, Standards and Scalable Agencies", In Proceedings of Autonomous Agents 2000 Workshop on Infrastructure for Scalable Multi-agent Systems, Spain

Chihiro Ono received the B.E degree of Electrical Engineering, the M.S degree of Computer Science from Keio University, Japan, in 1992 and 1994 respectively. Since joining KDD in 1994, he has been working on network management systems, database systems and agent systems for electronic commerce. From 1999 to 2000, he was a visiting researcher at Stanford University. He is currently a research engineer of Network Service Lab. in KDDI R&D Laboratories Inc.

Satoshi Nishiyama is a senior research engineer at Network Service Laboratory, KDDI R&D Laboratories, Inc. He received the B.E. degree of Electrical Engineering from The University of Tokyo and the M.A. degree of Computer Science from The University of Texas at Austin in 1984 and 1991 respectively. Since joining KDD (currently KDDI) in 1984, he has focused on the research on database design methodologies, data mining algorithm, matching algorithm for production systems, agent systems, transaction protocol, active networks and various issues on network management systems.

Keesoo Kim is a Ph.D. Candidate in Construction Engineering and Management Program, Department of Civil and Environmental Engineering, Stanford University. His Ph.D. research has been focusing on distributed coordination of project participants in construction projects using agent technologies. His research interests include application of trust information to construction supply-chain management. Prior to this, he has worked for Daewoo E&C Co, Ltd for nine years in domestic and international construction projects. He earned B.S. (1985) in Architectural Engineering from Seoul National University, Seoul, Korea, M.S. (1996) and Engineer (1997) in Civil and Environmental Engineering from Stanford University.

Boyd C. Paulson, Jr. is the Charles H. Leavell Professor of Engineering in Stanford's Graduate Program in Construction Engineering and Management. He was a Visiting Professor at the University of Tokyo, the Technical University of Munich, the University of Strathclyde in Glasgow, Scotland, and the University of Hawaii. He earned his

B.S., M.S., and Ph.D. degrees in Civil Engineering from Stanford University. He is the author of two books and over 100 papers. His research and teaching interests focus on computer applications in construction, construction equipment and methods, international construction, and affordable housing. He has had numerous sponsored research projects in these and other areas and has received several national and international awards for his work. He has studied construction projects and organizations in visits to numerous countries in six continents, including fellowships and grants to support research and teaching in China, Germany, Japan and the UK.

Mark Cutkosky is a principal investigator of the Design Interface for 3D manufacturing and Biometric Robotics at the Center for Design Research(CDR), director of the Dextrous Manipulation Lab, and co-director of the Stanford Alliance for Innovative Manufacturing at Stanford University. He joined the Design Division of the Stanford Mechanical Engineering Dept. in 1985 after working for several years in the Robotics Institute at Carnegie-Mellon University and as a machine design engineer at ALCOA, in Pittsburgh, Pennsylvania. He received B.S. of Mechanical Engineering at University of Rochester (1978), M.S(1982). and Ph.D.(1985). of Mechanical Engineering at Carnegie-Mellon University.

Charles J. Petrie, Jr. is the founding Executive Director of the Stanford Networking Research Center (SNRC), and a Sr. Research Scientist, Stanford Computer Science Department (pending). Prior to this, he was a Senior Research Scientist at the Center for Design Research (CDR) at Stanford University. He was the founding Editor-in-Chief of the IEEE Internet Computing journal. Prior to Stanford, he was a founding member of the MCC AI Lab where he led the Proteus project, which had the first commercial use of AI software from that lab, and founded the ongoing International Conference on Enterprise Modeling Technology (ICEMT). He has organized many workshops and conferences and is often an invited speaker. He has published extensively on inferencing architectures and the use of truth maintenance for concurrent engineering design, developing the seminal Redux model of design. He is co-developer of the widely used JATLite agent platform. His current research interest is the dynamic revision of web services using agent technologies. He has a Ph.D. degree in Computer Science from the University of Texas at Austin. He is a member of AAAI, IEEE Computer Society, and ACM.